UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/670,898 | 09/25/2003 | William J. Masek | LOT920030024US1 | 5987 |

45544 7590 12/10/2010
HOFFMAN WARNICK LLC
75 STATE ST
14TH FLOOR
ALBANY, NY 12207

| EXAMINER | |
|---|---|
| MITCHELL, JASON D | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2193 | |

| NOTIFICATION DATE | DELIVERY MODE |
|---|---|
| 12/10/2010 | ELECTRONIC |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

PTOCommunications@hoffmanwarnick.com

| | | | |
|---|---|---|---|
| ***Office Action Summary*** | **Application No.**<br>10/670,898 | **Applicant(s)**<br>MASEK ET AL. | |
| | **Examiner**<br>JASON MITCHELL | **Art Unit**<br>2193 | |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *20 August 2010*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-9,11-18 and 20-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9, 11-18, 20-26* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

## DETAILED ACTION

This action is in response to an amendment filed on 8/20/10.

Claims 1-9, 11-18 and 20-26 are pending in this application.

### *Response to Arguments*

In the par. on pg. 8, the applicants state:

> ... Applicants assert that the cited references fail to teach or suggest, inter alia, "instantiating, via the test script, a plurality of instances of the test application using threads, wherein the instantiating and execution of each of the plurality of instances of the test application occur within a single process." The Office admits that Duggan fails to teach or suggest that the plurality of instances occur within a single process, and instead asserts that this feature is taught by Dinker. The Office posits that Dinker teaches each test agent 110 may be a implemented as a multithreaded application, citing Par. 0028 and 0031 of Dinker, and that using the single agent of Duggan with the multithreaded agents of Dinker teaches the features of the claim. (Office Action, Pages 6-7). However, Applicants respectfully disagree with the combination of the references. For instance, as is clearly stated in Dinker, the test agents 110 are consistently a plurality of test agents, not a single agent. Dinker in fact teaches away from the single test agent of Duggan, stating that "by implementing test cluster 100 from several multi-threaded test processes (i.e. test agents 110)" there is "less thread starvation than if the same number of clients were simulated by a single multi-threaded process." (Dinker, Para. 0031). As such, Dinker viewed alone or in combination with Duggan fails to teach or suggest the feature of instantiating each of the plurality of instances of the test application within one process. Claims 9 and 18 include similar features. Firth and Partamian fail to cure this deficiency. Accordingly, Applicants respectfully request that the Office withdraw its rejection.

The examiner respectfully disagrees. The basis of the rejection is that those of ordinary skill in the art in attempting to implement the test environment of Duggan would be forced to determine how to implement Duggan's plurality of threads (e.g. col. 21, lines 53-61 'The basic module 12 is also responsible for initiating multiple, concurrent sessions ... Each session is executed as a separate thread"). Dinker teaches a method of initiating multiple, concurrent threads in which each of the plurality of threads is instantiated and executed within a single process (par. [0031] “multi-threaded test processes (i.e., test agents 110)”). The fact that Dinker teaches additional benefits which would have been achieved by spawning multiple processes (each containing multiple threads) would not have prevented those of ordinary skill in the art from implementing Duggan’s plurality of threads in a single process.

In the first par. on pg. 9, the applicants state:

> With further respect to independent claim 1, Applicants respectfully assert, in addition to the above arguments, that the cited references also fail to teach or suggest, inter alia, "sharing all services and memory space exclusively dedicated to the single process with others of the plurality of instances." Claim 1. The Office admits that Duggan and Dinker do not explicitly disclose sharing all services and memory space among the plurality of instances. Rather, the Office cites a passage of Partamian that teaches a general description of processes and threads, specifically citing that "threads in the same process share information using memory, atomic structures, mutexes, semaphores, etc." (Partamian, Col. 3, lines 12-17, emphasis added). Applicants, however, assert that the Office has misinterpreted the passage cited, as emphasized above. The passage states that threads within a process share information by using memory, not that they share a memory space. There is further no reference to sharing services. However, for clarification, Applicants have amended the claim to clearly indicate that the services and memory space are dedicated exclusively for the single process. As currently amended, Partamian clearly fails to teach or suggest this feature, as sharing information using memory clearly is not services and memory space dedicated for a single process. As such, the cited combination fails to teach or suggest that all services and memory space are shared among the plurality of instances, and Fink fails to cure this deficiency. Accordingly, Applicants request that the Office withdraw the rejection of claim 1. Further, claims 9 and 18 include similar features and withdrawal of the rejections is respectfully requested.

The examiner respectfully disagrees. While it is recognized that Partamian's phrasing differs from the claim language, those of ordinary skill in the art would have understood the teaching to meet the broadly worded claim. Specifically, by ‘using’ memory to share information Partamian is teaching an area of memory dedicated to the process for the purpose of sharing information amongst the threads. Further, those of ordinary skill in the art would have understood the reference to “atomic structures, mutexes, [and] semaphores" to describe structures which could reasonably be construed as the broadly claimed services and which are shared by the threads of the process. Even further, those of ordinary skill in the art would have understood that 'services' such as mutexes and semaphores describe tools used by the process to control access of the processes threads to other services such as processors (see e.g. the Wikipedia articles titled "Mutual Exclusion” and “Semaphore (programming)”).

In the par. bridging pp. 9 and 11, the applicants state:

> With further respect to independent claim 1, Applicants continue to respectfully assert, in addition to the above arguments, that the cited references also fail to teach or suggest, inter alia, "... identifying application protocol interfaces (APIs) ..., prior to the instantiating step... [and] providing a test script capable of invoking the APIs ...". Claim 1. The Office admits that Duggan, Dinker, and Partamian do not explicitly disclose that its command module is implemented as APIs. Rather, the Office continues to cite a passage of Firth that teaches, generically, that APIs exist, reciting "functions in the Internet API reside in a dynamic link library (DLL)." Col. 2, lines 63-67. First, this reference describing API simply describes it with no reference to test applications or any related information. Further, as previously stated, the Office attempts to replace whole pages of Duggan that describe the formation of scripts with one generic sentence describing where an API is stored. Although the Office attempts to explain that only one sentence of Duggan would be replaced (Office Action, Page 4), this is simply not true. Duggan's entire disclosure relies upon the use of Visual Basic 5.0. ... As such, entire pages do describe the use of Visual Basic, which is the Microsoft Program used in the entirety of the application. In fact, at Page 9 of the Office Action, the Office quotes the ease of use of Duggan with no knowledge of underlying programmed instruction of the command module needed, citing Col. 14, Lines 2-4 of Duggan. Applicants note that this is in reference to the preferred embodiment described in both Col. 13 and Col. 14, which utilizes the Visual Basic program for such an ease of use. Applicants respectfully submit that the reference to an API in Firth has nothing to do with creating testing of application

> programs, as in Duggan, and any attempt to incorporate this generic API into the Duggan Visual Basic GUI based system of Duggan would in fact, at best, lead to undue experimentation and yield unpredictable results. Accordingly, Applicants request that the Office withdraw the rejection of claim 1, as well as claims 9 and 18 which include similar features.

The examiner respectfully disagrees. The thrust of the applicants' argument is largely unclear. It appears that the applicants are asserting that use of the Visual Basic language (version 5.0) somehow precludes the development of APIs. Those of ordinary skill in the art would have understood that APIs (implemented e.g. as DLLs) could certainly have been written using Visual Basic 5.0. Further those of ordinary skill in the art would have realized that although Duggan makes reference to Visual Basic 5.0 the same functionality could have been defined using any number of different programming languages. Accordingly a generic teaching that APIs exist (e.g. Firth col. 2, lines 63-67 "functions in the Internet API reside in a dynamic link library (DLL)") is sufficient to show that implementing Duggan's test environment using the known APIs was within the ordinary level of skill in the art. In other words, meeting the claim language requires no more than saving Duggan's 'modules' in, for example, a .DLL file format instead of an .EXE file format. This is not the same as "replac[ing] whole pages of Duggan" (which would not be persuasive of non-obviousness regardless).

### *Claim Rejections - 35 USC § 112*

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

**Claims 1-9, 11-18 and 20-26 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement.** The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

**Claim 1** recites “a single process, sharing all services and memory space which are exclusively dedicated to the single process with other of the plurality of instances”. Par. [0017] of the specification states “multiple test applications will be running in the same process, sharing the same services and memory space”. This is the only reference to "sharing” services and memory space to be found in the disclosure. Nowhere does the disclosure explicitly describe services or memory “exclusive” and/or “dedicated” to a process. Accordingly, the newly added limitation (“which are exclusively dedicated to the single process") was not described in the specification in such away as to reasonably convey that the inventors had possession of the claimed invention at the time the application was filed.

**Claims 2-8** depend from claim 1 and are rejected accordingly.

**Claims 9, 11-18 and 20-26** make similar recitations or depend from claims that do and are rejected accordingly.

### *Claim Rejections - 35 USC § 103*

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

**Claims 1-5, 7-9, 11-14, 16-18, 20-23 and 25-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,002,871 to Duggan et al. (Duggan) in view of US 2004/0199815 A1 to Dinker et al. (Dinker) in view of US 7,062755 B2 to Partamian et al. (Partamian) in view of US 5,987,517 to Firth et al. (Firth).**

**Regarding Claims 1, 9 and 18:** Duggan discloses:

providing a test application that satisfies reentrancy requirements *(col. 21, lines 57-61 'Each session is … reentrant')* on a client *(col. 5, lines 18-21 'the test tool … runs on a single computer')*;

identifying command modules associated with the test application *(col. 12, lines 21-23 "A list box 272 contains a list of all of the commands in the command module created for testing a given application program", the command module is inherently identified to the list box in order for the list box to present all of the commands from that module; col. 14, lines 22-28 "the command module is implemented as a Visual Basic 5.0 code module, Each command of the command module comprises a Visual Basic subroutine that contains the instructions for the execution segment of the command")*;

providing a test script capable of invoking the command modules *(col. 13, lines 59-62 "a test operator [can] create test scripts containing … command module commands")*; and instantiating, via the test script, a plurality of instances of the test application using threads *(col. 21, lines 57-61 'Each session is executed as a separate thread'; col. 21, lines 46-50 "handles the execution of a test run based on a ... test script")*.

Duggan discloses instantiating and executing a plurality of instances of the test application under the control of a single application (*col. 21, lines 53-61 'The basic module 12 is also responsible for initiating multiple, concurrent sessions … Each session is executed as a separate thread … It is the multi-threaded, reentrant nature of the test tool program code"*). However, Duggan does not explicitly disclose the instantiating and execution of each of the plurality of instances of the test application occur within a single process, without requiring multiple processes to instantiate the plurality of instances within.

Dinker teaches a testing program which instantiates and executes each of a plurality of instances of the test application as one of a plurality of threads in a process (par. [0028] Each test agent 110 may be implemented as a multithreaded application"; par. [0031] "multi-threaded test processes (i.e., test agents 110)").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to instantiate and execute each of the plurality of instances of the test application (Duggan *col. 21, lines 53-61 'initiating multiple, concurrent sessions … Each session is executed as a separate thread"; Dinker par. [0028] "Each test agent 110 may* *be implemented as a multithreaded application"; par. [0031] "multi-threaded test processes (i.e., test agents 110)")* within a single process without requiring multiple processes to instantiate the plurality of instances within (e.g. by only implementing Duggan's single test agent 110 instead of Dinker's multiple test agents). Those of ordinary skill in the art would have been motivated to do so as one of a finite set of known and implementable methods of providing the disclosed functionality (*i.e. the threads are either implemented in the same process or different processes*) which would produce only the expected results (Duggan *col. 21, lines 53-61 'initiating multiple, concurrent sessions ... Each session is executed as a separate thread ... It is the multi-threaded, reentrant nature of the test tool program code"; par. [0031] "multi-threaded test processes (i.e., test agents 110)"*).

Duggen discloses the code of the individual threads can safely share services and memory space which are exclusively dedicated to the single process with the other threads (col. 21, lines 53-61 "each session is reentrant"). However, Duggen and Dinker do not explicitly teach sharing all services and memory space which are exclusively dedicated to the single process among the plurality of instances.

Partamian teaches sharing all services and memory space which are exclusively dedicated to the single process among the plurality of instances (*col. 3, lines 12-17 "A process may have one or a plurality of threads. Threads in the same process share information using memory, atomic instructions, mutexes, semaphores, etc."; note that* *this teaches "Threads in the same process" and does not indicate that threads in a different process have access to the services and memory space; further note that interprocess sharing mechanisms are distinctly described*).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to share all services and memory space (*Partamian col. 3, lines 12-17 "Threads in the same process share information using memory, atomic instructions, mutexes, semaphores, etc."*) among the plurality of threads (*Duggen col. 21, lines 53-61 'initiating multiple, concurrent sessions … Each session is executed as a separate thread … each session is reentrant"; par. [0031] "multi-threaded test processes (i.e., test agents 110)"*). Those of ordinary skill in the art would have been motivated to do so as one of a finite set of known and implementable method of providing the disclosed functionality (i.e. either the threads share memory and services or they don't) which would have resulted in only the expected results (*Duggen col. 21, lines 53-61 'initiating multiple, concurrent sessions … Each session is executed as a separate thread … each session is reentrant"; Partamian col. 3, lines 12-17 "Threads in the same process share information using memory, atomic instructions, mutexes, semaphores, etc."*).

Duggan, Dinker and Partamian do not explicitly teach the command module implemented as APIs.

Firth teaches the use of APIs *(col. 2, lines 63-67 "functions in the Internet API reside in a dynamic link library (DLL)*").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement Duggan's command module *(col. 14, lines 22-28 "the command module is implemented as a Visual Basic 5.0 code module)* as an API and to provide a data entry field in the GUI to identify particular API's for use with the application under test. Those of ordinary skill in the art would have been motivated to do so because Firth's APIs "eliminate the need to embed source code directly in an application program to manage Internet application protocols" *(col. 2, lines 64-67; also see Duggan col. 16, lines 9-15 "each command simulates a real user's interaction ... by generating ... an HTTP request")* and thus provide further abstraction for Duggan's test script development *(see e.g. col. 13, lines 59-67 "No knowledge of the underlying programmed instruction of the command module is needed by a test operator"; col. 14, lines 2-4 "The command module is rewritten and/or customized for each different application program to be tested").*

**Regarding Claim 2:** The rejection of claim 1 is incorporated; further Duggan discloses; and

upon execution, the test script instantiates the plurality of instances of the test application *(col. 5, line 67-col. 6, line 3 'the test tool program executes multiple concurrent sessions')* using threads *(col. 21, lines 57-61 'Each session is executed as a* *separate thread')* within the single process *(col. 21, lines 53-57 'The basic module 12 is also responsible for initiating multiple, concurrent sessions'; col. 21, lines 57 "It is the multi-threaded, reentrant nature of the test tool program code")*.

**Regarding Claims 3, 14 and 23:** The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan discloses the server application is a network application *(col. 5, lines 9-12 'a test tool for testing application programs … over a network')*.

**Regarding Claims 4, 12 and 21:** The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan discloses the reentrancy requirements dictates that the plurality of instances of the test application be run within the single process without interfering with each other *(col. 21, lines 57-61 'reentrant nature of the test tool')*.

**Regarding Claims 5, 13 and 22:** The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan discloses each of the plurality of instances of the test application corresponds to a separate thread *(col. 21, lines 57-61 'Each session is executed as a separate thread')*, and wherein each of the separate threads is associated with a different connection to the server *(col. 5, line 66-col. 6, line 3 'A "session" refers to the execution of one test script, on one client connection')*.

**Regarding Claims 7, 16 and 25:** The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan discloses the plurality of instances of the test application simulate use of the server application by a plurality of users *(col. 6, lines 47-51 'the test tool program … is capable of executing test scripts … based on a user list').*

**Regarding Claims 8, 17 and 26:** The method of claim 1, 9 and 18 further comprising collecting data corresponding to the test *(col. 8, lines 4-6 'The test tool program … provides four options for logging information').*

**Regarding Claims 11 and 20:** The rejection of claims 9, and 18 are incorporated respectively, further; Duggan discloses, and wherein upon execution, the test script instantiates the plurality of instances of the test application *(col. 5, line 67-col. 6, line 3 'the test tool program executes multiple concurrent sessions')* using threads *(col. 21, lines 57-61 'Each session is executed as a separate thread')* within the single process *(col. 21, lines 53-57 'The basic module 12 is also responsible for initiating multiple, concurrent sessions').*

**Claims 6, 15 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,002,871 to Duggan et al. (Duggan) in view of US 2004/0199825 A1 to Dinker et al. (Dinker) in view of US 7,062755 B2 to Partamian et al. (Partamian) in view of US 5,987,517 to Firth et al. (Firth) in view of "The Java $^{tm}$ Virtual Machine Specification" by Lindholm et al (Lindholm).**

**Regarding Claims 6, 15 and 24:** The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan does not disclose the process comprises a JAVA virtual machine.

Lindholm teaches that JAVA programs, which run on a JAVA virtual machine were well known at the time of the invention, and that JAVA programs and the JVM provided benefits known to those of ordinary skill in the art.

It would have been obvious to a person of ordinary skill in the art at the time of the invention to implement Duggan's 'test tool' and 'basic module' in the JAVA programming language and execute them on a JVM.

## *Conclusion*

**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JASON MITCHELL whose telephone number is (571)272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Bullock Lewis can be reached on (571) 272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jason Mitchell/
Primary Examiner, Art Unit 2193